福井大学
工学部　研究報告
第34巻　第2号
昭和61年9月

ラインエディタEDの改良について

柳瀬龍郎*　水野広治*

Improvement of the ED

Tatsuro YANASE and Hiroharu MIZUNO

(Received Aug 2, 1986)

The ED is a standard line-editor on the UNIX operating system. However, it is word processing oriented rather than character edit. We have improved on the line-editor mainly about following three points; 1) be able to change by a character pointed on any line, 2) can use memo pads, 3) multiple file edit. The program will be running on the new M-360 computer system in Fukui University.

1 初 め に

計算機システムで最も頻繁に利用され，そして恐らく，その計算機システムに対するイメージを決定的にしてしまうほど重要なシステムプログラムは編集用プログラム，いわゆるソースエディタであろうと思われる。ソースエディタとはすなわちプログラム，データ，文書やその他の様々な文字情報を作り出し変更するための道具となるものである。通常はメーカーお仕着せの，原始的な編集機能を持ったものを使う羽目になるのであるが，それでも利用者が文句も言わずに使っているのは，Kernighan〔1〕らの指摘を持つまでもなく，『自分達がどんなに損をしているか知らないから』である。

本報告では，上記の点を考慮しＵＮＩＸで利用されている高級なラインエディタＥＤの機能を増強し，これを実際にインプリメントした例について報告する。

＊　情報工学科

ＵＮＩＸは米国ＡＴ＆Ｔ社のベル研究所で開発されたオペレーティング・システムです。

2　ＥＤについて

現在，スクリーンエディット（画面編集）が全盛であるのに，なにゆえわざわざ，ラインエディット（行編集）なのかと言えば，それは主として次の２つの理由からである。

(1)　複雑なスクリーン制御コードを使わない。

(2)　ＥＤの文型探索機能が強力。

(1)の事柄から，汎用計算機のＴＳＳ端末として，専用端末ではなく各社のマイコンが使われるという，少なくとも福井大学内ではその傾向がうかがえる最近の状況に柔軟に対処できると思われる。また，専用端末によるスクリーンエディタの使い易さは論をまたないが，(2)に掲げた文型探索の機能が貧弱である。また，上の２つの他に，一般的入出力機能が備わっている，広域的な操作がシンプルなコマンド体系でサポートされているなどの理由があげられている。

さて，そのＥＤであるが，基本的には「行エディタ」であり，編集指令は常に１個または数個の行に対して作用する。もちろん行の一部に操作を加えることもできるようになっている。また，このエディタでは編集したいファイルの内容を直接操作せず，そのコピーをいったん作業ファイルにいれ，その作業ファイルを編集するようになっている。このため編集終了前に，書き出しコマンドによって，編集をおえた作業ファイルを意識的に元のファイルの上に重ね書きするか，もしくは，新しいファイル名をつけて書き出ししないかぎり，元のファイルの内容は，決して変更されず，また編集した結果を残すことができない。コマンドの実行は作業ファイルに対してのみ行われるのである。以下の各節においてＥＤの特徴をみることにする。

2.1　使用文字

このエディタでは英小文字での指令も可能である。その他，特別扱いの文字として，次の様なメタ文字が使われている。

？……㊎をのぞく任意の文字

％……行の先行

＄……行の末尾，またはファイルの最終行

＊……閉包，直前の文型の０回以上の繰り返し

@……脱出記号

＆……繰り返し記号

／……行内操作コマンド区切り記号

@Ｎ……㊎

@Ｔ……ＴＡＢ記号

2.2　コマンドの形成

エディタへの入力はコマンドの列であり，コマンドは一行には１個しか書けない。各コマンド行は，広域指定部を設けない場合，次のような形をしている。

行１，行２　指令　追加情報　㊎

「行１」，「行２」および一部の「追加情報」は省略可能である。

2.3　広域指定部

広域指定部は指令の実行を複数の行にわたって繰り返させる役をする。指令はＧ指定のときは文型に一致した各行について，またＸ指定のときは文型に一致しなかった各行について，それぞれ実

行される。形式は次の通り。〔以下においては（×××）は×××を省略してもよいことを表す〕

(n1(,n2))G／文型／指令	文型に一致した各行について
(n1(,n2))X／文型／指令	文型に一致しなかった各行について
↑ 広域指定部	n1，n2を省略すれば，1，$として解釈

2.4 行番号について

ファイルの行番号は先頭行から相対的に自然数が対応ずけられており，行レコードに絶対的に付属しているものではない。したがって挿入や削除によってその都度変化する。また，コマンドの前にある行番号は，コマンドを実施すべきファイルの中の範囲を指定する意味を持つ（範囲の両端を含む）。コマンドにおける行番号は次のような構成要素から作られる。

17	10進数
.（ピリオド）	現在の行
$	最後の行
／文型／	順方向の文型サーチ
¥文型¥	逆方向の文型サーチ

ここで，文型サーチを行番号の指定に用いるという意味は，発見された文型を含む行の番号をその値とすると言う意味である。また，これらの構成要素は＋または－で結合できる。例えば，

.＋1	.と1の和
.－2	.と2の差

行番号はコンマ（，）またはセミコロン（；）で区切られる。セミコロンがくると先へ進むに先だって，点.（ピリオド）にそのセミコロンの直前にあらわれた行番号の値が代入される。コマンドの先頭には任意の個数の行番号を付けることができる（但し一部のコマンドを除く）。必要に応じて，これらの行番号のうちの最後の一つまたは二つが使われる。行番号が二つ必要なとき，一つしか与えられなかったならば，その1個の行番号が両方の行番号として使われる。行番号が一つも与えられていないときには，次に示す省略時解釈のいずれかが使われる。例えば，

(.)	現在の行を使う
(.＋1)	次の行を使う
(.－1)	前の行を使う
(.,.)	現在の行を二つの行番号として使う
(1,,$)	全部の行を使う

等である。それぞれのコマンドについての省略時解釈は，デフォルト（既定値）としてあらかじめ定められている。

2.5 行操作コマンド

行操作コマンドは行単位での編集ができるように作られており，行追加，行挿入，行変更，行複写，行移動，行書き出し，行削除，行復活および行表示等のコマンドがある。

2.6 文型操作

指定した文型を含む行を探したり，ある文型を指定した文字列で置き換えたりすることができるが，ＥＤでは特にこの文型として使用できる文字列パターンに特徴があるので，次に詳しくのべる。

2.6.1 文型について

文型は文型探査（サーチ），置換指令S，および広域指定部Gとでつかわれ，つぎのものからなる。

c	文字そのもの
?	改行符号を除く任意の文字
%	行の先頭
$	行の先頭（改行符号の直前の空列）
[・・・]	文字の類（これらの文字の内の任意の一つ）
[_・・・]	否定（アンダーライン）付きの文字以外なら何でも
	例 [_A－Z]
*	閉包（直前の文型の0回以上の繰り返し）
@c	脱出記号付きの文字（@%，@[，@*，@N（㊋），@／など）

すなわち，文型は形式言語理論における一重カッコの正規表現（従ってサブクラス）が許され，しかも文脈自由で探索可能である。また，文字が持つ特別の意味は脱出記号が付けられたとき，[・・・]，[_・・・]の中，および次の場合には失われる。

*	が先頭にあるとき
$	が末尾以外にあるとき
%	が先頭以外にあるとき

文字の類はつぎのような要素を0個以上ならべて[ ]で囲んだものである。

c	文字そのもの。[でもよい
a－c	文字の範囲（数字，下段の英字，上段の英字）
@c	脱出記号付きの文字（@_，@－，@@，@]）
_	先頭にあれば文字の類の否定を表す（_A，_c－d）

文字の類の中にあらわれる文字の特別な意味は，脱出記号が付いているときおよび次の場合には失われる。

－	が先頭または末尾にあるとき
_	が先頭以外にあるとき

置き換え文字列は次の要素を0個以上並べたものから成る。

c	文字そのもの
&	繰り返し記号。すなわち任意の一致した文字列
@c	脱出記号付きの文字（@&）

脱出記号付きの文字は文字@のあとに1個の文字を付けたものから成る。

@N	は改行符号 ㊋
@T	は欄とばし符号（TAB）
@c	はcそのもの（@@をふくむ）
	をあらわす。

2.6.2 文型操作コマンド

ある文型を探してそれを別の文字列で置き換えるときに使用されるコマンドで，このコマンドに

おいてサーチする文型を指定した場合，その文型は記憶される。また文型を省略した場合，記憶されている文型がサーチに使用される。文型変換コマンドは

（n1(，n2))S／(文型1)／文字列2／(G)　　(2)

の形をしている。コマンドの意味は，n1からn2までサーチして文型1を文字列2で置き換える。行を指定しなかった場合，現在行が操作の対象となる。追加情報のGを指定した時，その一行中の該当する全ての文型が文字列2によって置き換えられる。文字列2の任意の位置では，@N（NL改行符号）の使用も許される。従って，@N以外の文字を，@Nで置き換えれば一行が分割されて複数行になる。またPが指定されたときは変換をうけた最後の行が表示される。そのほか，単に文型のサーチをすることも可能で，そのときには，

前方サーチ・・・"／(文型)／"

後方サーチ・・・"¥(文型)¥"

で，その文型を含む行を探すことができる。

2.7 ファイル操作

ファイル操作コマンドとしては，ファイル読み込みコマンドRにより，ファイルの内容を，nまたは現在行の後に続いて読み込む事ができる。そのほか，ファイル書き出しコマンドWによって任意の行を書き出してファイルを作ることもできる。また，内部バッファを空にしてから，編集を行う新ファイルの内容をバッファに読み込むコマンド，どんなファイル名が記憶されているかを見るコマンド，ファイルのレコード内容を表示するコマンド等がある。

2.8 その他コマンド

コマンドの使い方の簡単な表示をおこなうヘルプコマンド，メッセージの表示を制御するコマンド等がある。

表1に，EDの本来のコマンドとその機能を掲げておく。

表1　ED本来のコマンドとその機能

A		行の次へ文を付加（あとに文が続く）
C		文を変更（あとに文が続く）
D		文を削除
E	ファイル名	ファイルの取り込み
F	ファイル名	ファイル名を印刷
G		広域指定部
H		ヘルプ表示
I		文を行の前に挿入（あとに文が続く）
M		文を移動
P		文を表示
NL		次行表示（NLのみ入力）
Q		終了
R	ファイル名	ファイルの読み込み（行のあとへ付加）
S		文型置き換え
U		削除取り止め
W	ファイル名	ファイルの書き出し（状態は不変）
X		広域指定部
=		行番号表示

3　追加された機能

ＥＤは高級なラインエディタではあるが，欠点がないわけではない。特にＥＤに欠けている機能として，１行内における編集機能があげられる。すなわち，ある文型を含む行を探索する機能は強力であり，かつ使い易いが，その行の望む部分を自由に指定し編集する機能が弱い。また，最近のエディタで装備されている，複数ファイル編集機能や，メモリ機能などもＥＤには欠けている。そこで，表２に示す様な機能を新しく追加した。

表２　新しく追加した機能

〔機　　能〕	〔コマンド〕
1. 行内編集	（ｎ１（，ｎ２））：
2. メモリへの書き出し	（ｎ１（，ｎ２））Ｂ（ｎ３）
3. メモリからの読み出し	（ｎ１）ＲＲ（ｎ２）
4. 編集バッファの切り換え	＃（Ｅ　ファイル名）
5. エディトバッファの表示	（ｎ１（ｎ２））Ｔ
ファイルの表示	Ｔ（ｎ１（，ｎ２））　（ファイル名）
メモリの表示	ＴＴ［ｎ］
6. コマンドファイルの実行	！＜コマンドファイル名＞

ただし，表２において，ｎはメモリ番号，（ ）は省略可能である。以下において，これらの機能を詳細に見てゆく。

3.1　行内編集機能

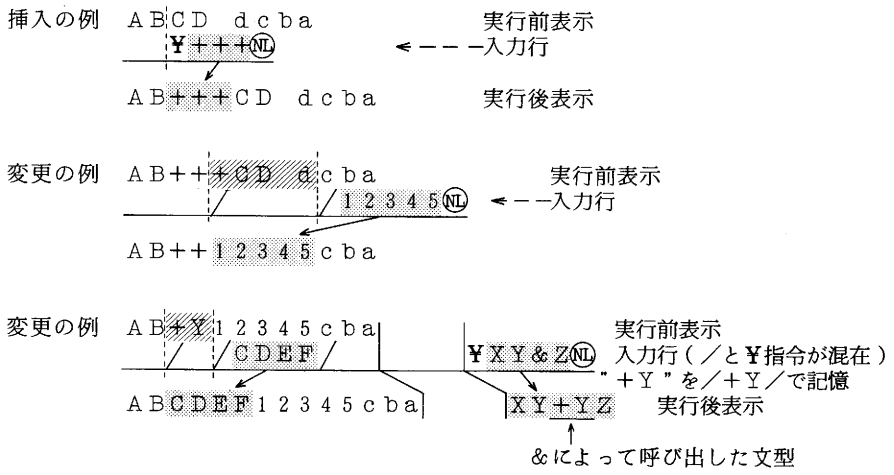

図１　行内編集例

コマンドは次のような形式となる。

（ｎ１（，ｎ２））：　　（３）

最近のスクリーンエディタの考え方だけを取り入れ，カーソルによって，編集する位置を指定する方法である。ただし，図１に示すように，カーソルは表示された被編集行の下をスペースとデリートＫＥＹによって，左右にのみ動かすようになっている。そして，挿入する場合は挿入する位置

の後の文字の下で挿入記号"¥"を入力し，変更または削除する場合は，削除する文字列の先頭の文字と，最後の文字の次の文字の下へ削除記号"／"を書き，その後へ，新しい文字列を入力して最後にコマンド区切り記号"／"，または，改行を入力する。変更された行はそのつど表示しなおされ，改行入力のみとなるまで「表示」と「修正入力」の反復を行う。特徴としては，入力する文字列の中に，ある定められた方法で，「改行文字」，「ＴＡＢ」などを入力することができることである。すると，この「改行文字」を入力したところで，元の文字列は切断されることになる。複数行を一行につなぐのは別のコマンドで行う。

3.2　メモリと編集バッファに関する機能

最近のエディタでは複数個のファイルが同時に扱えるようになっている。また，編集における所謂，切貼を簡単にするための機能を取り入れた。

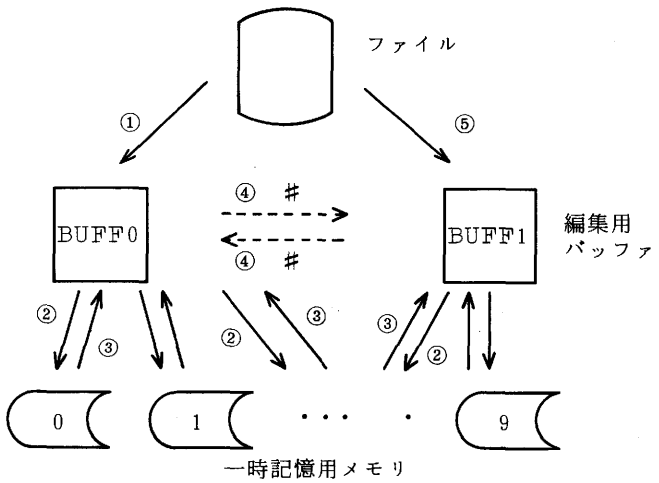

図２　ファイル，編集用バッファ，一時記憶用メモリの相互関係

図２に，外部ファイル，編集バッファおよびメモリの関係を，データの動きと共に示してある。実線矢印がファイルや，テキストの流れを表し，その側にその動きを引き起こす，コマンドが示してある。また，破線矢印は，編集の対象となるバッファの変更を表している。すなわち，①では，コマンド

（ｎ１）　Ｒ　（ｎ２（，ｎ３））（ファイル名）　　（４）

によって，ファイル名で示した外部ファイルのｎ２行からｎ３行までを，編集中のｎ１行の後ろへ，追加挿入することができることを示している。このときｎ１を省略すれば，現在行の直後に挿入が行われる。ｎ２，ｎ３を共に省略するとファイル全体が読み込まれ，ｎ３を省略するとｎ２行以降の全ての内容が続み込まれる。またファイル名を省略すると，記憶されている既定値が使用される。このときファイル名の直前には，必ず１個以上空白がなければならない。なお，このコマンドはもともと一本のファイルの全てを読み込むようになっていたものを，一部だけ読めるように修正した。

②では，メモリへ書き出しを行っており，

（ｎ１（，ｎ２））Ｂ（！）（ｎ３）　　（５）

によって，編集中のｎ１行からｎ２行までがメモリｎ３へ書き出される。このとき，以前から記憶

されている内容の後ろへ追加される。"！"があれば以前の内容は全てクリアされる。また，

(1) ｎ１，ｎ２を共に省略した場合，現在行のみが書き出され，

(2) ｎ２を省略した場合，第ｎ１行のみが書き出され，

(3) ｎ３は０～９の値をとることによって10個のメモリのいずれかを指定することができ，省略した場合，自動的に０が指定される。

③では，メモリからの読み出しを行い，

（ｎ１）ＲＲ（ｎ２）　　（６）

でｎ１行の直後へ，メモリｎ２の全ての内容を読み出して，挿入を行う。

(1) ｎ１を省略すると，現在行の後へ挿入が行われ，

(2) ｎ２を省略すると，自動的にメモリ０が選ばれて読み出しが行われる。

④では，編集バッファの切換を行い，

＃　　（７）

で二つの編集バッファを交互に往来できる。二つのバッファでのデータのやりとりは，メモリを介して行う。二つの編集バッファ⑤は，バッファの切換えと同時に，編集ファイルを指定しており，

！Ｅ（ファイル名）

で，新しい編集バッファを用意して（ファイル名）で指定されたファイルをそこで編集することを指示している。＃を省略すれば，現在の編集バッファを全くクリアして，すなわち，何かあっても全てご破算にして，ファイル名で指定したファイルを新しくセットし直すことを示す。いずれの場合も，ファイル名を省略すれば，すでにシステムで記憶されたファイル名が既定値として使用される。バッファの数，すなわち，同時編集できるファイルの数は，現在のところ二つに制限してあるが，論理的にはほとんど制限はなく，パラメータの数を１つ変えるだけで変更可能なように設計してある（前述のメモリ数についても全く同じ）。

3.3 表示機能

Ｔで任意のファイルの任意の行，または，編集バッファの任意の行を表示，ＴＴで任意のメモリの内容全てを表示することが可能である。すなわち，

(1) テキストの表示　（ｎ）Ｔ　編集テキストのｎ行から最終行まで表示。

(2) ファイルの表示　Ｔ（ｎ１（，ｎ２））（ファイル名）
ファイルのｎ１行からｎ２行まで表示，ｎ２を省略すると最終行まで，ｎ１，ｎ２とも省略すれば，全ての行が行番号付きで表示される。ファイル名については他の場合と同様。

(3) メモリ内容の表示　ＴＴ（ｎ）　メモリは０から９まで10個あり，どのメモリを表示するかをｎで指定する。ｎを省略すると０が選択される。メモリの内容は全て表示される。

いずれの場合も，何らかのＫＥＹを打つことによって表示は中止される。

3.4 コマンドファイル実行機能

エディタを起動し，編集を行っている途中で，長いコマンドを打ち込む場合や，同じコマンドを繰り返したい場合などに，ファイルからコマンド入力を行うことができる。

やり方は簡単で，一担，そのコマンド列を内容とする一つのファイルを作る。そしてそのファイル名の先頭に記号＂！＂を付けて入力する。

！ファイル名　　（９）

すると，エディタへのコマンドは，全て，そのファイルから入力されて実行され，ファイル中の全てのコマンドを実行した後，また，キーボードからの入力持ちになる。コマンドファイルの作成は編集バッファを切換えて行ってもよいし，また，現在編集中のファイルの最後から最初までのラインの外側で作って書出しコマンドで，名前を付けて書き出し，その後，その行を消去するなど，方法はいくつか存在する。

４　ま　と　め

ＵＮＩＸ上で動くラインエディタを改良しコマンドを追加して，使い易くした例を報告した。追加したおもな機能は，行内編集機能，複数ファイル操作機能，一時記憶機能，コマンドファイル実行機能，等である。大形コンピュータのＴＳＳを利用する際に，専用端末ではなく，手許にある使い慣れたマイコンやパソコンを使いたい場合がかなりある。そんなとき，各種のスクリーン制御コードを使うスクリーンエディタよりも，制御コードの簡単なラインエディタのほうが，端末として使われる装置を選ばない。そのラインエディタそのものを改良して使い易くすれば，端末としてのマイコンやパソコンもそれなりに用途が拡がることになり，ホストの計算機システムも有効に利用されることにもなる。現在，ＦＯＲＴＲＡＮとＣで書かれた二つのバージョンが稼動している。

参　考　文　献

1) Brain W. Kernighan and P.J.Plauger : "Software Tools" ADDISON-WESLEY, 1981
（日本語訳　「ソフトウエア作法」木村　泉訳　共立出版）